Chapter Six: Contents

(Selectors/Iteration Databases– 14 July 2001 – LA-UR 00-1725 – TRANSIMS 2.1)

1. INTRODUCTION 1

1.1 OVERVIEW.......... 1

1.2 THE ITERATIVE PROCESS 4

1.3 TRANSIMS FRAMEWORK FLEXIBILITY 7

2. ALGORITHM.......... 9

2.1 OVERVIEW.......... 9

2.2 COLLATOR.......... 9

2.3 STRATIFIER.......... 11

2.4 SELECTOR 12

APPENDIX A: ITERATION DATABASE GENERAL CONFIGURATION FILE KEYS 13

APPENDIX B: ITERATION DATABASE ACTIVITY CONFIGURATION FILE KEYS 15

APPENDIX C: ITERATION DATABASE MICROSIMULATION EVENT CONFIGURATION FILE KEYS 18

APPENDIX D: ITERATION DATABASE ROUTER/PLAN CONFIGURATION FILE KEYS..........20

APPENDIX E: ITERATION DATABASE POPULATION CONFIGURATION FILE KEYS 22

APPENDIX F: ITERATION DATABASE STRATIFIER CONFIGURATION FILE KEYS 23

APPENDIX G: ITERATION DATABASE ALGORITHM CONFIGURATION FILE KEYS..........25

CHAPTER SIX: INDEX 29

Chapter Six: Figures

Fig. 1. The selection process. 3

Fig. 2. Location of the Selector/Iteration Database within a typical TRANSIMS experimental design. 5

Fig. 3. Typical Selector/Iteration Database logic. 5

Fig. 4. Four examples of iteration progressions. 6

Fig. 5. Typical Selector/Iteration Database data flow. 7

Chapter Six—Selectors/Iteration Databases

1. INTRODUCTION

1.1 Overview

A key distinguishing feature of TRANSIMS is the process known as iterative feedback. Feedback provides a natural way to tailor models (of activity locations, mode selections, route planning, etc.) to specific, possibly overlapping, subpopulations. Feedback enables the overall computational system to reflect “learned” behavior within the simulated population represented. Feedback involves two crucial processes:

- Biased selection – defining a subpopulation based on any static or dynamic information about travelers available to TRANSIMS.
- Updating travelers – revising the selected subpopulation’s use of the transportation system by controlling the quality of information about the system available to them.

The information about travelers available to TRANSIMS consists of the traveler-specific data contained in population, activity, plan, vehicle, and simulation output files. These data are all generated by TRANSIMS under specific hypotheses about the transportation network. By carefully controlling the hypotheses, TRANSIMS can be used to steer travelers toward certain choices.

The mechanics of controlling information flow among TRANSIMS modules is discussed in the Input/Output section of each module’s description. This chapter describes the Selector/Iteration Database and how it works together with an iteration script to control the overall TRANSIMS Framework. A typical TRANSIMS study involves repeated iteration between modules. There is no single, “standard” iteration script because different study designs involve different iteration schemes.

One important example of feedback is in solving the traffic assignment problem. The simplest version of this uses a loop between the Route Planner and Traffic Microsimulator modules. On the first pass of the Route Planner, routes are chosen under the hypothesis that travel time is well represented by free speeds on the network (i.e., that travelers do not interact). Correction for traveler interactions can be applied simply by making available to the Route Planner information about actual travel times produced by the Traffic Microsimulator[1]. With this information, the Route Planner will choose different routes for most travelers, resulting in different travel times. In this case, updating travelers is accomplished by re-running the Route Planner with an updated travel time table. However, there is still a wide range of different feedback schemes for this problem which depend on the selection step—exactly which travelers are to be run through the Route Planner with updated travel time information. One selection process is to choose a certain fraction of travelers uniformly at random. The Selector/Iteration

[1] Notice that there is no requirement to provide *correct* travel time information—it might be noisy, or averaged together with travel times used in the previous run.

Database described below supports much more sophisticated processes, though. For example, one could select only travelers with automobile drives of an hour or more who cross a geographic feature (like a river).

Of course, there are many more information flows in TRANSIMS than just the travel time table. Every TRANSIMS module can be used to update only a selected subpopulation using information provided by the Framework. In effect, this is like providing a separate model for every conceivable subdivision of the population without the need for fitting each model separately. For example, work location is chosen using a single simple model for the entire population. If people who commute by bus across a river are assigned work locations poorly, selecting that subpopulation and running the work location assignment model with slightly different input information can change the poorly selected locations for that subpopulation with no change in the model itself.

Notice that a single traveler might be in *two* subpopulations – for example, the previous subpopulation and the subpopulation assigned to households larger than five people who also have longer than average commutes—but no sophisticated correlation structure needs to be built into the model to handle such cases correctly.

Selection is based on both absolute criteria such as traveler’s mode, and on relative criteria such as the duration of a trip compared to the duration of all other trips in the subpopulation picked out by the absolute criteria. The relative criteria act as user-specified cost functions. Thus, we might select the 10% of travelers meeting some absolute criteria who have the longest actual travel time compared with their expected travel time.

Fig. 1 gives an indication of how the selection process works. Here data is collected on the travelers’ incomes, their travel modes, the length of trips, whether they cross the river, and the relative length of the trip. All travelers with some collection of these characteristics, for example those on bus trips with income >$40k, are collected and the distribution of relative trip duration is formed. A portion of these travelers with the largest duration is selected to travel by a different mode.

Selection and Feedback

The iteration database:

Traveler	Income	Mode	>1 hour?	Cross river?	Relative duration	. . .
291362	$25K	bus	no	yes	1.2	. . .
291363	$34K	car	yes	no	1.6	. . .
291364	$42K	car	no	yes	1.1	. . .
291365	$ 0K	walk	no	no	1.0	. . .
291366	$38K	car	yes	yes	2.3	. . .
291367	$45K	bus	yes	no	1.4	. . .
291368	$30K	car	yes	yes	1.3	. . .

Selection criterion:	Selects travelers:
bus trips with income >$40K	291367
short trips crossing the river	291362 291364
long car trips not crossing the river, relative duration > 1.3	291363

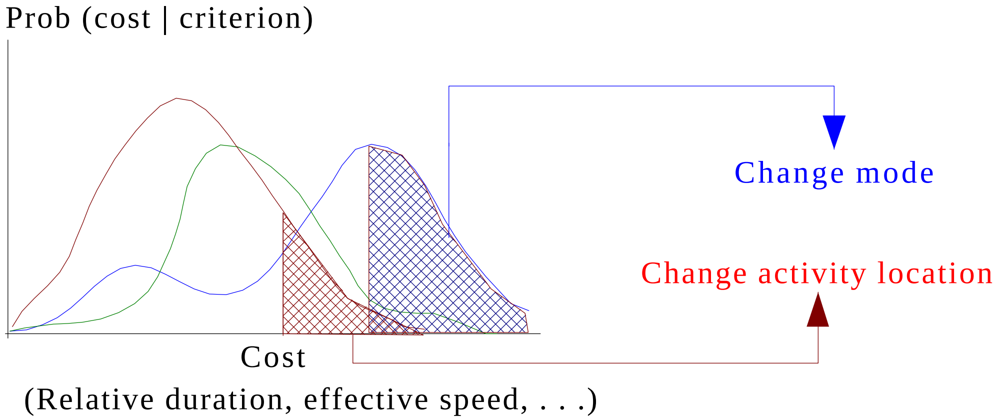

Fig. 1. The selection process.

1.2 The Iterative Process

Users can prepare an iteration script to control the entire iteration process. The script uses special control commands specifically developed for this iteration of TRANSIMS components. It enables the user to filter results, run repeated iterations, establish stopping criteria, and perform a host of other operations that make the analyst's job less labor intensive.

During each iteration, the iteration script controlling the current study typically invokes a Selector/Iteration Database. (The script might even use a different Selector/Iteration Database for each iteration in a study.) When a Selector runs, it usually will do the following:

- Read information about the travelers from the Iteration Database.
- Examine each traveler and decide whether to
    - regenerate his or her activities using the Activity Generator,
    - select a new route between his or her existing activities using the Route Planner, or
    - retain his or her existing activities and the planned route between them.
- Write the selections made for each traveler into data files that can be read by the Activity Generator and the Route Planner when they are executed.
- Summarize the selections made and the current state of the system into a Selector/Iteration Database statistics data file.

Fig. 2 and Fig. 3 illustrate a Selector/Iteration Database's decision-making process.

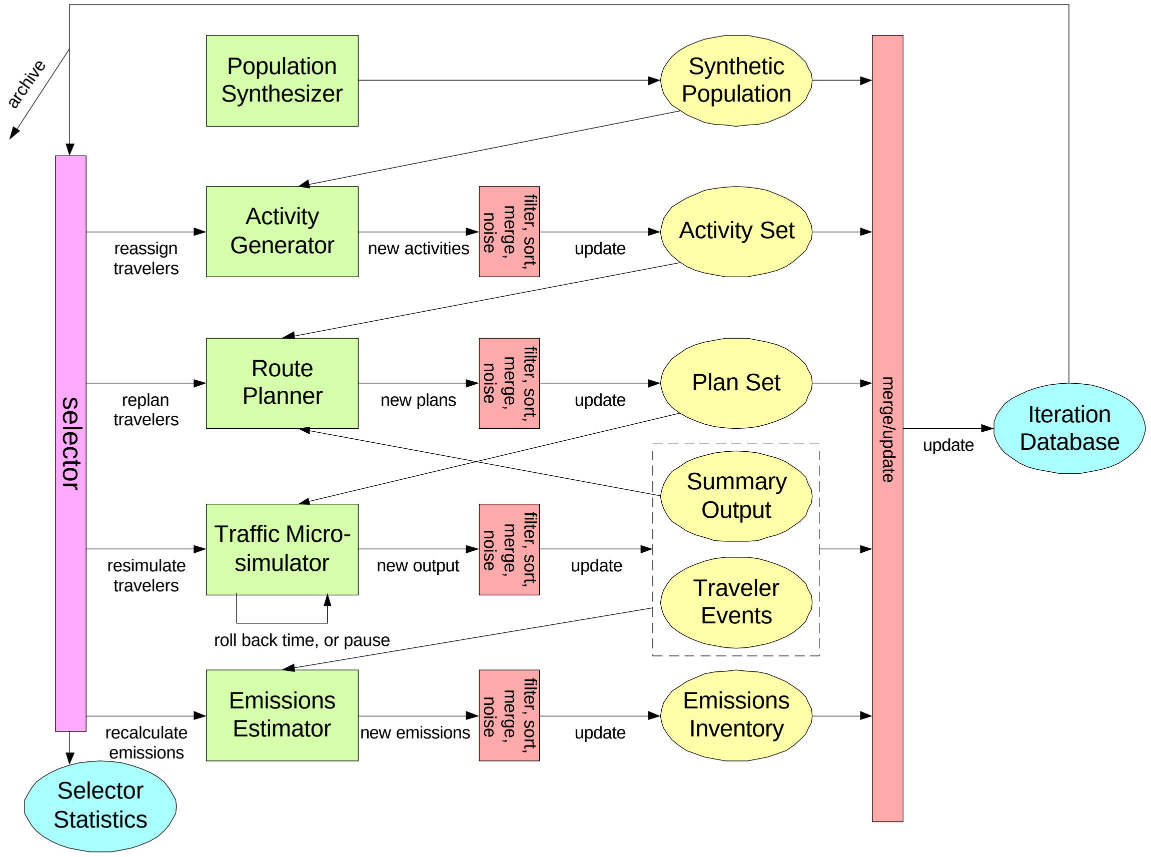

Fig. 2. Location of the Selector/Iteration Database within a typical TRANSIMS experimental design.

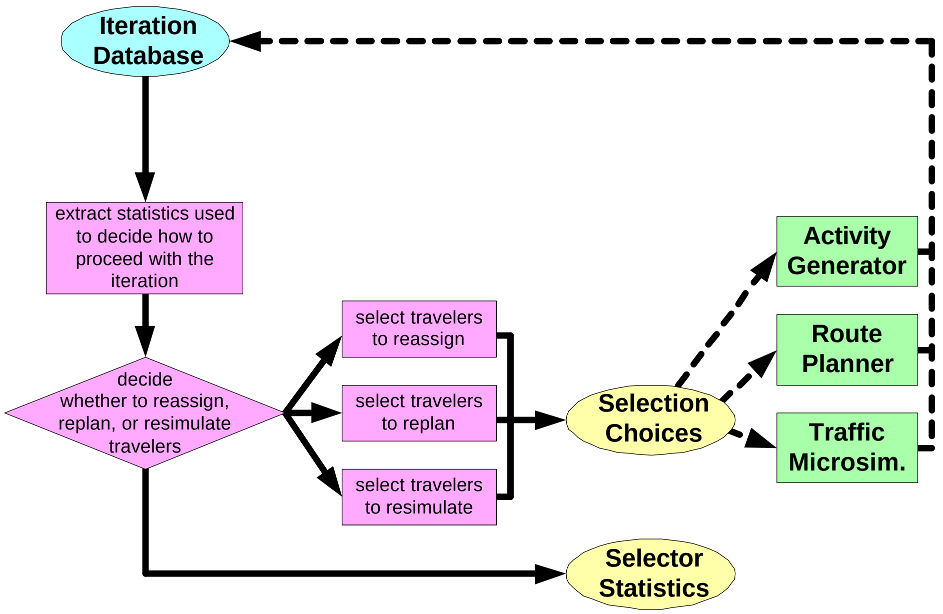
Iteration Database
extract statistics used to decide how to proceed with the iteration
decide whether to reassign, replan, or resimulate travelers
select travelers to reassign
select travelers to replan
select travelers to resimulate
Selection Choices
Activity Generator
Route Planner
Traffic Microsim.
Selector Statistics

Fig. 3. Typical Selector/Iteration Database logic.

After the Selector completes the selection process for all travelers, the Activity Generator, Route Planner, or Traffic Microsimulator runs to calculate the updated activity set, plan set, or microsimulation output files, respectively (according to the decisions made by the Selector).

The iteration script will reinvoke a Selector again at the start of the next iteration in the study. Fig. 4 shows examples of four possible progressions, as determined by the Selector.

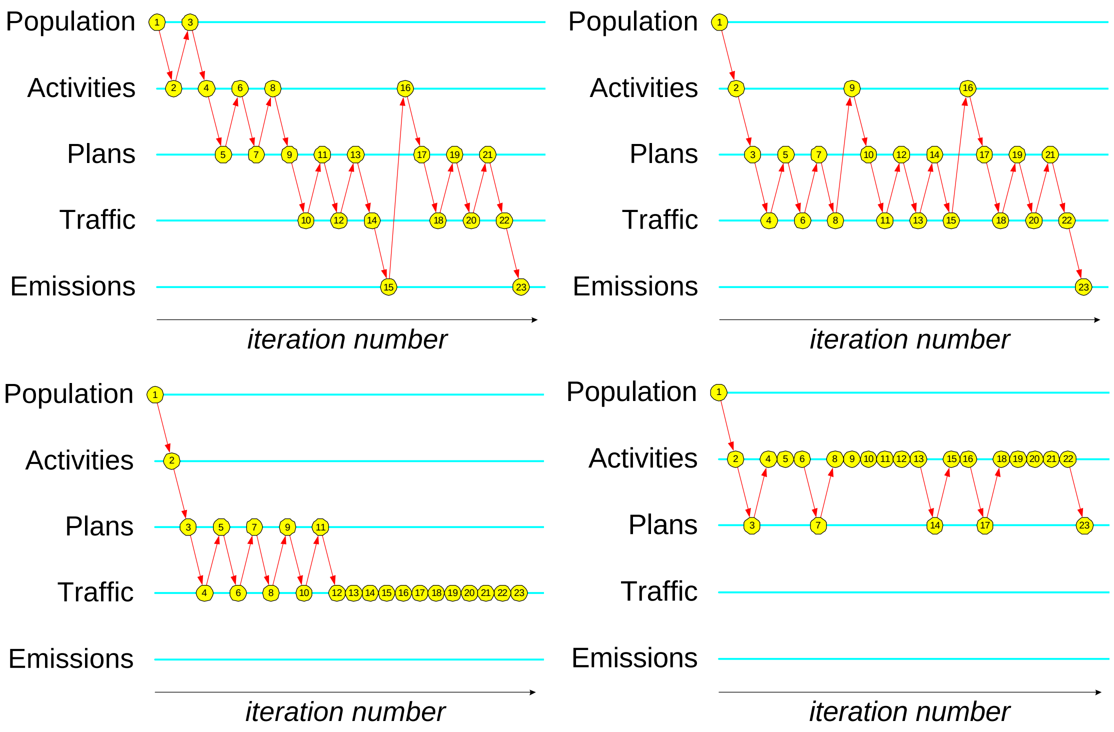

Fig. 4. Four examples of iteration progressions.

The Iteration Database is the archive of information about travelers across iterations. The Selector uses this information to make its selection decisions. The data contained in the database are chosen by the user from:

- The fields of the population, activity, and plan files—for example, income, mode preference, or the expected duration of a trip.
- Information extracted from detailed Traffic Microsimulator event output—for example, the actual duration of a trip.
- Information deduced from combinations of the previous two—for example the duration of a trip relative to its expected duration.

The left side of Fig. 5 shows this data flow into the Selector/Iteration Database.

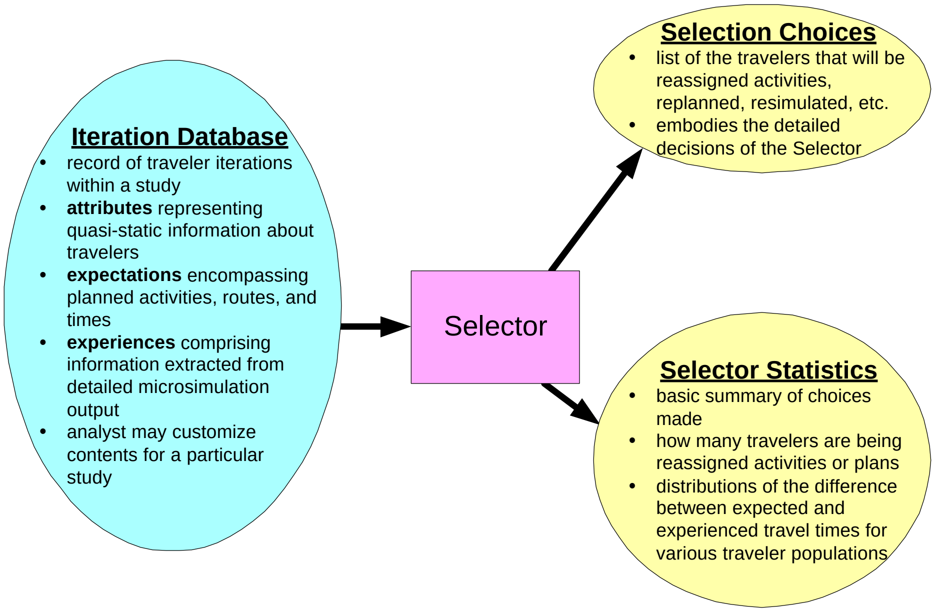
Iteration Database
record of traveler iterations within a study
attributes representing quasi-static information about travelers
expectations encompassing planned activities, routes, and times
experiences comprising information extracted from detailed microsimulation output
analyst may customize contents for a particular study
Selector
Selection Choices
list of the travelers that will be reassigned activities, replanned, resimulated, etc.
embodies the detailed decisions of the Selector
Selector Statistics
basic summary of choices made
how many travelers are being reassigned activities or plans
distributions of the difference between expected and experienced travel times for various traveler populations

Fig. 5. Typical Selector/Iteration Database data flow.

The Selector has two principal outputs: Selector/Iteration Database statistics and selection choices. The selection choice files simply list the travelers that will be reassigned activities or will be re-planned, re-simulated, etc. These files embody the Selector/Iteration Database's detailed decisions.

Selector/Iteration Database statistics provide a basic summary of the choices a Selector/Iteration Database makes (e.g., how many travelers are being re-planned, distributions of the difference between expected travel times, and experienced travel times for various traveler populations). The right side of Fig. 5 shows this data flow out from the Selector/Iteration Database.

1.3 TRANSIMS Framework Flexibility

The Framework's flexibility allows for countless variations in the iteration process. For example, in some studies, the Selector may run after the Activity Generator or Route Planner completes its execution. Thus, the Selector can decide which of the generated activities or plans will be accepted for travelers. Those not accepted are discarded and new activities or plans are produced.

Users can also design Selectors that will feed travelers to the Activity Generator or Route Planner (one-by-one) so that the Selector, Activity Generator, Route Planner, and Traffic Microsimulator all execute simultaneously, with their coordination controlled by the Selector. This may increase the computational efficiency of a study, thus allowing for new experimental designs with finely controlled iteration.

The iteration script has the potential to make additional choices, such as the following:

- which version of the Activity Generator, Route Planner, or Traffic Microsimulator will run during the present iteration;
- if transit schedules will be adjusted or vehicles added or removed from the transit fleet;
- if network characteristics (such as traffic signal timing, congestion pricing, or roadway information signs) will be altered;
- which travelers receive data from traffic information systems; or
- whether to complete the study (i.e., end the iteration) because the iterations have converged sufficiently (or diverged).

Several implementations have been written that have use in typical transportation planning studies. For example, Fig. 3 shows a typical iteration scheme that is set up by the Selector/Iteration Database script. In this scheme, activities, plans, and microsimulations are iterated until traffic behavior on the network stabilizes. It is not difficult for analysts to write additional Selectors/Iteration Databases for their own specialized studies.

2. ALGORITHM

2.1 Overview

This section specifies the intended interface to a suite of tools providing Selector/Iteration Database functionality. These tools are known as the Collator, Stratifier, and Selector.

2.2 Collator

Each module of TRANSIMS provides information that can be used to fill in some of the fields of the Iteration Database. For example, after running the Population Synthesizer, demographic information can be collected; after running the Route Planner, expected travel times can be collected. The Collator can be run after each module and will fill in all of the fields in the Iteration Database that depend on that module with the most recent data available.

As its name indicates, the Collator's main function is to gather data from disparate sources (e.g., activity files, plan files, event files) into a single table keyed by traveler ID and trip number. The user specifies which data are to be collected using configuration file keys of the form `SEL_USE_<name>`, where `<name>` is the name of a particular piece of data. See the configuration file key section for information on the possible values for `<name>`. In addition, the Collator will accumulate data over an entire trip and provide some commonly used processing algorithms described below.

Every record of the Iteration Database includes the following identifying information:

- `HH` – the household ID
- `TRAV` – the traveler ID
- `TOUR` – an integer identifying which home—home tour this trip belongs to
- `SUBTOUR` – an integer identifying which round trip from a non-home anchor location this trip belongs to (0 if it is not part of a subtour)
- `TRIP` – the trip ID integer
- `START_ACT_ID` – the ID of the starting activity location for this trip
- `END_ACT_ID`, the ID of the ending activity location for this trip

The records are stored in a comma-delimited ASCII text file. This arrangement makes it easy for an analyst to use familiar statistical analysis tools on the data as well as the analysis tools provided by TRANSIMS. For example, a simple text processing tool might be used to create a single record for each tour a traveler makes, summarizing the trip-based data records provided by the Collator. Similarly, an analyst might wish to extract the total travel time for each traveler on each iteration and build a cross-iteration database.

Because the Collator must know the traveler ID and trip for each piece of data it collects, Traffic Microsimulator event output files must include the TRAVELER and TRIP fields if they are to be used by the Collator.

2.2.1 Processing Algorithms

One form of processing determines whether a traveler starts or ends in a user-defined region or moves from one region to another. This processing is triggered if the configuration file contains the SEL_USE_START_IN_REGION, SEL_USE_END_IN_REGION, or SEL_USE_CROSS_BOUNDARY keys, respectively.

The SEL_USE_START_IN_REGION and SEL_USE_END_IN_REGION configuration file keys' values should be a semicolon-separated list of user analysis zones (UAZ), which are defined by the user. UAZs may be a coarsening of traffic analysis zones (TAZ), but they may also be completely different. The UAZs are defined in separate files of the form used to specify regions for the Output Visualizer. That is, the first line gives the number of polygons to follow. Each line after that gives a polygon (UAZ) ID followed by the number of vertices in the polygon, then by the x,y,z values for each vertex. The interpretation of UAZs will ignore the “z” value.

The coordinate system will be the one used by the network node table. There could be more than one set of UAZs: one might be used to define the Central Business District, for example, while another distinguishes the two sides of a river. There will be a sequentially numbered set of UAZs, one for each UAZ_FILE_n key. The first value of n must be 1. Defining a region for these two keys requires a pair of integers: the first is the value of n specifying the desired UAZ file; the second is the polygon ID to use. For each such pair (n,m), the Iteration Database will include a field labeled START_IN_REGION_n_m (or END_IN_REGION_n_m).

The value of the SEL_USE_CROSS_BOUND configuration file key is just the n specifying what UAZ file to use. For each n specified, there will be a separate field in the Iteration Database labeled CROSS_BOUND_n. This field will be true if the beginning and ending activity locations are in different polygons. For the purposes of this algorithm, any point not contained in one of the given polygons is assumed to lie in an undeclared polygon. For example, a UAZ file containing a single polygon would tacitly define two regions: the inside and the outside of the polygon.

Another useful piece of data calculated by the Collator is the Euclidean distance between the start and end activity locations for each trip. This will be calculated if the SEL_USE_EUCLID configuration file key is set. It is also possible to extract information from the user-defined fields of the Activity Location Table for the starting and/or ending locations. For example, if the user has included a traffic analysis zone field in the Activity Location Table, The Collator can be configured to add the start and end TAZ zone number to the Iteration Database with the following lines in the configuration file:

```
SEL_USE_START_ACT_USER_DATA_TAZ
```

```
SEL_USE_END_ACT_USER_DATA_TAZ
```

The Collator will summarize several aspects of a traveler's trip. It will determine whether the traveler drives a passenger on any leg if the configuration file key `SEL_USE_DRIVES_PASSENGER` is set. It will write a string describing the transportation mode used on each leg (e.g., wcwtc) if `SEL_USE_MODE_STRING` is set. It will count the number of legs in each trip if `SEL_USE_NUMLEGS` is set. Furthermore, it will count the number of legs with a particular mode if `SEL_USE_MODE_LEG_COUNT` is set to the mode(s) of interest. The configuration file keys `SEL_USE_T_WALK`, `SEL_USE_T_MODE`, `SEL_USE_T_WAIT`, and `SEL_USE_T_TOTAL` trigger the Collator to calculate the amount of time during a trip spent walking, in the requested mode, waiting, or in total, respectively. These times are derived from the plan files where available, or from the Traffic Microsimulator output event files. Also from this file come the effective speed (Euclidean distance / total time, triggered by `SEL_USE_EFFSPEED`) and a flag that is true only for trips that were completed in the Traffic Microsimulator (`SEL_USE_FINISH_TRIP`).

Another form of processing that the Collator does is to calculate the following simple functions of pairs of columns:

```
SUM (A, B) = A + B
DIFF (A, B)     = A - B
REL_DIFF (A, B) = (A - B) / B
PRODUCT (A, B) = A * B
RATIO (A, B)    = A / B
AND (A, B) = logical AND of (A, B)
OR (A, B) = logical OR of (A, B)
XOR (A, B) = logical Exclusive OR of A, B)
```

These functions can be specified using configuration file keys of the form `ITDB_USE_DIFF  <A>, <B>; <C>, <D>` , where <A>, <B>, etc., are the names of fields in the ITDB. (`REL_DIFF` could be calculated using `DIFF` and `RATIO`, but it is such a common operation that TRANSIMS provides a simple means of doing it.)

2.3 Stratifier

The Stratifier uses a combination of built-in algorithms on the information contained in the ITDB to stratify or classify trips. The classification is stored in the ITDB as indexes into a set of n-way user-specified tables. The tables themselves can be reconstructed from the configuration file used with the Stratifier.

The first step in specifying a stratification is defining a discretization, or binning, for each variable. Each binning is given a user-defined name, using the configuration file key `SEL_BIN_NAME_<n>`. Any raw or processed data field in the Iteration Database can be binned. The field is specified using the configuration file key `SEL_BIN_FIELD_<n>`. The boundaries of the bins will be determined automatically as quantiles of the data if the user specifies SEL_BIN_NUMBINS_<n>; or they may be specified explicitly by the user using the key SEL_BIN_BOUNDS_<n>; or (for data that is already discrete) each different value can be placed in a different bin. Open-ended bins extending to positive or negative infinity are automatically implied.

The Stratifier combines k different binnings specified using the configuration file key SEL_START_BINS into a single k-way table. It assigns an index to each different cell in the table. That index is assigned to every trip in that cell and also added to the corresponding field of the Iteration Database.

2.4 Selector

A Selector uses the Iteration Database to select a set of travelers. Each algorithm can be coded as a separate executable, or could be incorporated into a single monolithic Selector executable. A Selector algorithm has a name, a goal, and a cost function (optionally) associated with it. These attributes are defined by configuration file keys SEL_ALGORITHM_n, SEL_GOAL_n, and SEL_COST_n, respectively, where n and m are replaced by positive integers. The name is user defined, but must be unique across algorithms. Each algorithm may require its own set of parameters, which are passed along with the name. The goal is either to reroute the traveler or to make use of one of the feedback pathways defined in the Activity Generator.

The cost function must be the name of a field present in the Iteration Database when the Selector is run.

The Selector first associates each trip with the value of a cost function (chosen using the configuration file key SEL_COST from among the other fields of the Iteration Database). The distribution of cost values for all of the trips in a given cell can be used to select a cell for further processing. For example, the user might wish to find the cell with the highest variation in costs, or the lowest mean cost. The algorithm used to select a cell is given by the SEL_ALGORITHM configuration file key.

Once a cell has been chosen according to the user-specified algorithm applied to the user-specified cost function, there are several ways to select travelers from within the cell. For example, the user might select at random from all travelers within the cell, or from one of the tails of the distribution of costs. The algorithm used to select from within a cell is determined by the configuration file key SEL_BIN_SEL_ALGO.

Appendix A: Iteration Database General Configuration File Keys

Configuration File Key	Description
ROUTER_IGNORABLE_PRIORITIES	See the Route Planner documentation.
ACT_HOME_ACTIVITY_TYPE	The number of the home activity type (non-negative integer). This key should be specified if an activity file is specified.
ACT_SCHOOL_ACTIVITY_TYPE	The number of the school activity type (non-negative integer). This key should be specified if an activity file is specified.
ACT_WORK_ACTIVITY_TYPE	The number of the work activity (non-negative integer). This key should be specified if an activity file is specified.
NET_ACTIVITY_LOCATION_TABLE	The activity location table name. This key is required.
NET_DIRECTORY	The directory where the network files reside. This key is required.
NET_LINK_TABLE	The link table name. This key is required.
NET_NODE_TABLE	The node table name. This key is required.
NET_PARKING_TABLE	The parking table name. This key is required.
NET_PROCESS_LINK_TABLE	The process link table name. This key is required.
NET_TRANSIT_STOP_TABLE	The transit stop table name. This key is required.
SEL_ACTIVITY_FILE	The activity file for use by the Collator. If not present, the `ACTIVITY_FILE` configuration file key is used. Either `SEL_ACTIVITY_FILE` or `ACTIVITY_FILE` must be specified.
SEL_EVENT_FILE	The event file for use by the Collator.
SEL_HOUSEHOLD_FILE	The name of a file containing a list of household IDs from the synthetic population for which data will be collected in the Iteration Database. This configuration file key is optional and if not specified, data for all households in the population will be collected.
SEL_ITDB_FILE	The full pathname of the output Iteration Database file generated by the Selector Collator. An iteration number extension is automatically added to the end of this name.
SEL_MESSAGE_LEVEL	Sets the message level for the Collator, stratifier, and Selector modules. Message levels range from –1 to 4 with the higher numbers reporting more.
SEL_NO_ITDB_INDEX	If set to 1, the Collator does not create a traveler-based binary index for the Iteration Database. TRANSIMS modules currently do not use the index, which may take a considerable amount of execution time to produce. Default value is 0, which causes the index to be produced.
SEL_PLAN_FILE	The plan file for use by the Collator. If not present, the `PLAN_FILE` configuration file key is used.

Configuration File Key	Description
`SEL_POPULATION_FILE`	The population file for use by the Collator. If not present, the `ACT_POPULATION_FILE` configuration file key is used. Either `SEL_POPULATION_FILE` or `ACT_POPULATION_FILE` must be specified.
`SEL_STRAT_OUT_FILE`	The full pathname for the output Iteration Database created by the Stratifier. There may be several of these for each Collator run. Default = `strat`
`SEL_UAZ_FILE_n`	The full pathname of the polygon file specifying User Analysis Zone(s) `n`, where `n` is an integer starting at 1.

Appendix B: Iteration Database Activity Configuration File Keys

Configuration File Key	Description
`SEL_USE_ACT_HH_ID`	If set, directs the Collator to add the household ID from the activity file for the ending activity for the trip to the output Iteration Database
`SEL_USE_ACT_LAST_OK_ITER`	If set, directs the Collator to add a field containing the number of the last iteration on which no problems were reported for this household in the Activity Generator or Regenerator's Problem File. A value of `–1` indicates there has been a problem reported on every iteration.
`SEL_USE_ACT_PERSON_ID`	If set, directs the Collator to add the person ID from the activity file for the ending activity for the trip to the output Iteration Database
`SEL_USE_END_ACT_GROUP_NUM`	If set, directs the Collator to add the activity group number from the activity file for the ending activity for the trip to the output Iteration Database
`SEL_USE_END_ACT_ID`	If set, directs the Collator to add the activity ID from the activity file for the starting activity for the trip to the output Iteration Database
`SEL_USE_END_ACT_LOCATION`	If set, directs the Collator to add the first of the possible locations from the activity file for the ending activity for the trip to the output Iteration Database
`SEL_USE_END_ACT_TYPE`	If set, directs the Collator to add the activity type from the activity file for the ending activity for the trip to the output Iteration Database
`SEL_USE_END_DUR_LB`	If set, directs the Collator to add the duration lower bound from the activity file for the ending activity for the trip to the output Iteration Database. The value is converted from fractional hours to seconds.
`SEL_USE_END_DUR_UB`	If set, directs the Collator to add the duration upper bound from the activity file for the ending activity for the trip to the output Iteration Database. The value is converted from fractional hours to seconds.
`SEL_USE_END_TIME_LB`	If set, directs the Collator to add the end time lower bound from the activity file for the ending activity for the trip to the output Iteration Database.
`SEL_USE_END_TIME_UB`	If set, directs the Collator to add the end time upper bound from the activity file for the ending activity for the trip to the output Iteration Database.

Configuration File Key	Description
SEL_USE_END_MODE_PREF	If set, directs the Collator to add the mode preference from the activity file for the ending activity for the trip to the output Iteration Database
SEL_USE_END_OTHER_PARTICIPANTS	If set, directs the Collator to add the number of other participants from the activity file for the ending activity for the trip to the output Iteration Database
SEL_USE_END_POSS_LOC	If set, directs the Collator to add the number of possible locations from the activity file for the ending activity for the trip to the output Iteration Database
SEL_USE_END_PRIORITY	If set, directs the Collator to add the activity priority from the activity file for the ending activity for the trip to the output Iteration Database
SEL_USE_END_START_TIME_LB	If set, directs the Collator to add the start time lower bound from the activity file for the ending activity for the trip to the output Iteration Database
SEL_USE_END_START_TIME_UB	If set, directs the Collator to add the start time upper bound from the activity file for the ending activity for the trip to the output Iteration Database
SEL_USE_END_VEHICLE_ID	If set, directs the Collator to add the vehicle ID from the activity file for the ending activity for the trip to the output Iteration Database
SEL_USE_START_ACT_GROUP_NUM	If set, directs the Collator to add the activity group number from the activity file for the starting activity for the trip to the output Iteration Database.
SEL_USE_START_ACT_ID	If set, directs the Collator to add the activity ID from the activity file for the starting activity for the trip to the output Iteration Database.
SEL_USE_START_ACT_LOCATION	If set, directs the Collator to add the first of the possible locations from the activity file for the starting activity for the trip to the output Iteration Database.
SEL_USE_START_ACT_TYPE	If set, directs the Collator to add the activity type from the activity file for the starting activity for the trip to the output Iteration Database.
SEL_USE_START_DUR_LB	If set, directs the Collator to add the duration lower bound from the activity file for the starting activity for the trip to the output Iteration Database. The value is converted from fractional hours to seconds.
SEL_USE_START_DUR_UB	If set, directs the Collator to add the duration upper bound from the activity file for the starting activity for the trip to the output Iteration Database. The value is converted from fractional hours to seconds.

Configuration File Key	Description
`SEL_USE_START_END_TIME_LB`	If set, directs the Collator to add the end time lower bound from the activity file for the starting activity for the trip to the output Iteration Database. The value is converted from fractional hours to seconds.
`SEL_USE_START_END_TIME_UB`	If set, directs the Collator to add the end time upper bound from the activity file for the starting activity for the trip to the output Iteration Database. The value is converted from fractional hours to seconds.
`SEL_USE_START_MODE_PREF`	If set, directs the Collator to add the mode preference from the activity file for the starting activity for the trip to the output Iteration Database.
`SEL_USE_START_OTHER_PARTICIPANTS`	If set, directs the Collator to add the number of other participants from the activity file for the starting activity for the trip to the output Iteration Database.
`SEL_USE_START_POSS_LOC`	If set, directs the Collator to add the number of possible locations from the activity file for the starting activity for the trip to the output Iteration Database.
`SEL_USE_START_PRIORITY`	If set, directs the Collator to add the activity priority from the activity file for the ending activity for the trip to the output Iteration Database.
`SEL_USE_START_START_TIME_LB`	If set, directs the Collator to add the start time lower bound from the activity file for the starting activity for the trip to the output Iteration Database.
`SEL_USE_START_START_TIME_UB`	If set, directs the Collator to add the start time upper bound from the activity file for the starting activity for the trip to the output Iteration Database.
`SEL_USE_START_VEHICLE_ID`	If set, directs the Collator to add the vehicle ID from the activity file for the starting activity for the trip to the output Iteration Database.

Appendix C: Iteration Database Microsimulation Event Configuration File Keys

Configuration File Key	Description
SEL_USE_ACCELS	If set, directs the Collator to add the ACCELS field from the Traffic Microsimulator event output data to the output Iteration Database. This value is summed across events from every leg of the trip.
SEL_USE_ANOMALY	If set, directs the Collator to add the ANOMALY field from the Traffic Microsimulator event output data to the output Iteration Database. Note that this gives only the value for the last event of the trip.
SEL_USE_DISTANCE_SUM	If set, directs the Collator to add the DISTANCE_SUM field from the Traffic Microsimulator event output data to the output Iteration Database. This value is summed across events from every leg of the trip.
SEL_USE_EVENT_LEG_ID	If set, directs the Collator to add the LEG field from the Traffic Microsimulator event output data to the output Iteration Database. Only the value for the first event of each trip is reported.
SEL_USE_EVENT_PERSON_ID	If set, directs the Collator to add the TRAVELER field from the Traffic Microsimulator event output data to the output Iteration Database.
SEL_USE_EVENT_TRIP_ID	If set, directs the Collator to add the TRIP field from the Traffic Microsimulator event output data to the output Iteration Database.
SEL_USE_EVENT_USER	If set, directs the Collator to add the USER field from the Traffic Microsimulator event output data to the output Iteration Database.
SEL_USE_EVENT_VEHICLE_ID	If set, directs the Collator to add the VEHICLE field from the Traffic Microsimulator event output data to the output Iteration Database. Only the value for the first event of each trip is reported.
SEL_USE_LINK	If set, directs the Collator to add the LINK field from the Traffic Microsimulator event output data to the output Iteration Database. Note that this gives only the value for the last event of the trip.
SEL_USE_LOCATION	If set, directs the Collator to add the LOCATION field from the Traffic Microsimulator event output data to the output Iteration Database. Note that this gives only the value for the last event of the trip.
SEL_USE_NODE	If set, directs the Collator to add the NODE field from the Traffic Microsimulator event output data to the output Iteration Database. Note that this gives only the value for the last event of the trip.

Configuration File Key	Description
`SEL_USE_ROUTE`	If set, directs the Collator to add the `ROUTE` field from the Traffic Microsimulator event output data to the output Iteration Database. Only the value for the first event of each trip is reported.
`SEL_USE_SIGNALS`	If set, directs the Collator to add the `SIGNALS` field from the Traffic Microsimulator event output data to the output Iteration Database. This value is summed across events from every leg of the trip.
`SEL_USE_STATUS`	If set, directs the Collator to add the `STATUS` field from the Traffic Microsimulator event output data to the output Iteration Database. Note that this gives only the value for the last event of the trip.
`SEL_USE_STOPPED`	If set, directs the Collator to add the `STOPPED` field from the Traffic Microsimulator event output data to the output Iteration Database. This value is summed across events from every leg of the trip.
`SEL_USE_STOPS`	If set, directs the Collator to add the `STOPS` field from the Traffic Microsimulator event output data to the output Iteration Database. This value is summed across events from every leg of the trip.
`SEL_USE_TIME`	If set, directs the Collator to add the `TIME` field from the Traffic Microsimulator event output data to the output Iteration Database. Only the value for the last event of each trip is reported.
`SEL_USE_TIME_SUM`	If set, directs the Collator to add the `TIME_SUM` field from the Traffic Microsimulator event output data to the output Iteration Database. This value is summed across events from every leg of the trip.
`SEL_USE_TURN`	If set, directs the Collator to add the `TURN` field from the Traffic Microsimulator event output data to the output Iteration Database. Note that this gives only the value for the last event of the trip.
`SEL_USE_VEH_SUBTYPE`	If set, directs the Collator to add the `VSUBTYPE` field from the Traffic Microsimulator event output data to the output Iteration Database. Only the value for the first event of each trip is reported.
`SEL_USE_VEH_TYPE`	If set, directs the Collator to add the `VEHTYPE` field from the Traffic Microsimulator event output data to the output Iteration Database.
`SEL_USE_YIELDS`	If set, directs the Collator to add the `YIELDS` field from the Traffic Microsimulator event output data to the output Iteration Database. This value is summed across events from every leg of the trip.

Appendix D: Iteration Database Router/Plan Configuration File Keys

Configuration File Key	Description
`SEL_USE_COST`	If set, directs the Collator to add the cost from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the sum over all legs.
`SEL_USE_DEP_TIME`	If set, directs the Collator to add the departure time from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the first leg.
`SEL_USE_DRIVER`	If set, directs the Collator to add the driver flag from the Plan file for the trip to the output Iteration Database. The Collator will fill this field with `NA`, since there are multiple possible values per trip.
`SEL_USE_DURATION`	If set, directs the Collator to add the (expected) duration from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the sum over all legs.
`SEL_USE_END_ACC`	If set, directs the Collator to add the ending accessory ID from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the last leg.
`SEL_USE_END_ACC_TYPE`	If set, directs the Collator to add the ending accessory type from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the last leg.
`SEL_USE_GCF`	If set, directs the Collator to add the generalized cost function from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the sum over all legs.
`SEL_USE_LEG_ID`	If set, directs the Collator to add the leg ID from the Plan file for the trip to the output Iteration Database. The Collator will fill this field with `NA`, since there are multiple legs for each trip.
`SEL_USE_MAX_TIME`	If set, directs the Collator to add the max time flag from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the last leg.

Configuration File Key	Description
`SEL_USE_MODE`	If set, directs the Collator to add the mode from the Plan file for the trip to the output Iteration Database. The Collator will fill this field with `NA`, since there are multiple possible values per trip.
`SEL_USE_PLAN_PERSON_ID`	If set, directs the Collator to add the person ID from the Plan file for the trip to the output Iteration Database.
`SEL_USE_PLAN_USER`	If set, directs the Collator to add the User Field from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the first leg.
`SEL_USE_ROUTER_LAST_OK_ITER`	If set, directs the Collator to add a field containing the number of the last iteration on which no problems were reported for this traveler in the Router’s Problem File. A value of `–1` indicates there has been a problem reported on every iteration.
`SEL_USE_START_ACC`	If set, directs the Collator to add the starting accessory from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the first leg.
`SEL_USE_START_ACC_TYPE`	If set, directs the Collator to add the starting accessory type from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the first leg.
`SEL_USE_STOP_TIME`	If set, directs the Collator to add the (expected) arrival time from the Plan file for the trip to the output Iteration Database. Note that there are multiple legs in a trip, each of which contains a value for this field. The Collator will use the last leg.
`SEL_USE_TRIP_ID`	If set, directs the Collator to add the value in the user field from the Plan file for the trip to the output Iteration Database.

Appendix E: Iteration Database Population Configuration File Keys

Configuration File Key	Description
`SEL_USE_<pop_file header field>`	If set, directs the Collator to add the corresponding demographic variable from the population file to the output Iteration Database.
`SEL_USE_BLOCK_GROUP`	If set, directs the Collator to add the `BLOCK_GROUP` field from the population file to the output Iteration Database.
`SEL_USE_HH_ID`	If set, directs the Collator to add the `HH_ID` field from the population file to the output Iteration Database.
`SEL_USE_HOME_LOCATION`	If set, directs the Collator to add the `HOME_LOCATION` field from the population file to the output Iteration Database.
`SEL_USE_NUMBER_PERSONS`	If set, directs the Collator to add the `NUMBER_PERSONS` field from the population file to the output Iteration Database.
`SEL_USE_NUMBER_VEHICLES`	If set, directs the Collator to add the `NUMBER_VEHICLES` field from the population file to the output Iteration Database.
`SEL_USE_PERSON_ID`	If set, directs the Collator to add the `PERSON_ID` field from the population file to the output Iteration Database.
`SEL_USE_TRACT`	If set, directs the Collator to add the `TRACT` field from the population file to the output Iteration Database.

Appendix F: Iteration Database Stratifier Configuration File Keys

Configuration File Key	Description
SEL_ALGORITHM	A comma- or semicolon-separated list of names of algorithms used to select travelers. Currently, the allowed values are: HI_VAR, LO_VAR, HI_RANGE, LO_RANGE, HI_MEAN, LO_MEAN, HI_SIGMA, and LO_SIGMA. These values instruct the Selector to pick the cell with the highest or lowest variance, range, mean, or ratio of mean to standard deviation, respectively. In addition, it is possible to pick a cell by index using the argument PICK_CELL <n>, where <n> is an integer. If the k binnings that make up a cell have n_0, n_1, …, n_k bins each, a set of indexes into each binning of the form (i_0, i_1, …, i_k) is equivalent to the cell index $i_k + n_{(k-1)} * (i_{(k-1)} + \ldots + * (i_2 + n_1 * (i_0))\ldots)$. One algorithm must be supplied for each name in the SEL_COST argument.
SEL_BIN_BOUNDS_n	If specified, the bin boundaries used for binning n. The argument is a comma- or semicolon-separated list of values. Bins will be created extending from the smallest value to -infinity and from the largest value to + infinity. Over-ridden if SEL_BINS_NUMBINS_n is present.
SEL_BIN_FIELD_n	The column name of the input Iteration Database used for creating binning n.
SEL_BIN_NAME_n	The column name for binning n in the output Iteration Database. n must start at 0.
SEL_BIN_NUMBINS_n	If specified, the number of bins to create for binning n. Bin boundaries will be chosen to create bins with equal numbers of elements. The number of bins may be adjusted within the code if the distribution of elements is concentrated on a few values.
SEL_BIN_SEL_ALGO	A semicolon-separated list of names of algorithms to use in selecting trips from within the chosen cell of the stratification. Possible values are: RANDOM, TAIL, or ALL. There must be one algorithm supplied for each name in the SEL_COST argument. The RANDOM value takes two optional parameters: the first is the fraction of elements to select; the second is the absolute number of elements to select, which overrides the first if it is smaller. For example: RANDOM, 0.2, 100; TAIL takes one required and three optional arguments. The first is 0 if the lowest cost tail is to be selected and non-zero otherwise; the second is the fraction of trips to select; the third is an absolute threshold to apply; and the last is an absolute number of elements to select.
SEL_COST	A comma- or semicolon-separated list of names of columns in the input Iteration Database to associate with stratifications for use by the Selector in selecting trips. One selected set will be created for each name.

Configuration File Key	Description
`SEL_GOAL`	The Activity Regenerator command to be associated with the selected set of trips. One goal must be supplied for each name in the `SEL_COST` argument. The entire goal string is written to the Activity Generator feedback file after each selected traveler ID.
`SEL_STRAT_BINS`	A semicolon-separated list of comma-separated strings specifying the names of binnings in the Iteration Database to be used in stratifying the data.
`SEL_USE_STRATIFICATION`	Each semi-colon separated list in the value of `SEL_STRAT_BINS` creates one stratification, indexed beginning with 0. One stratification must be supplied for each name in the `SEL_COST` argument.

Appendix G: Iteration Database Algorithm Configuration File Keys

Configuration File Key	Description
SEL_USE_AND	Directs the Collator to include a field (for each pair) which is the logical AND between the values of the two fields. The argument is a semicolon separated list of comma-separated Iteration Database field names.
SEL_USE_CROSS_BOUND	Adds a field to the output Iteration Database which is true if the starting and ending activity locations for the trip are in different polygons. The argument is a comma-separated list of UAZ ID, as above, but without the polygon identifier. The value of this field is true if the starting and ending activity locations for the trip are in different polygons.
SEL_USE_DIFF	Directs the Collator to include a field (for each pair) which is the difference between the values of the two fields. The argument is a semicolon-separated list of comma-separated Iteration Database field names.
SEL_USE_DRIVES_PASSENGER	Directs the Collator to include a field which is true if, on any leg of the trip, the Plan file specifies that the traveler drives a vehicle with passengers. Does not apply to transit vehicle drivers.
SEL_USE_EFFSPEED	Directs the Collator to include a field giving the ratio of the Euclidean distance between start and end activities to the total time (as calculated for SEL_USE_T_TOTAL).
SEL_USE_END_ACT_USER_DATA	Directs the Collator to include a field giving the value of any user-specified field in the Activity Location network table for the activity location at the end of the trip. The argument is a semicolon-separated list of field names.
SEL_USE_END_IN_REGION	See SEL_USE_START_IN_REGION. The value of this field will be true if the ending activity location is inside the polygon.
SEL_USE_END_REGION	The number (n) of the User Analysis Zone specified by the configuration file key SEL_UAZ_FILE_n. The value of the field in the iteration database is the number of the polygon in the UAZ that contains the ending activity location or –1 if not in any defined polygon.
SEL_USE_EUCLID	Directs the Collator to include a field giving the Euclidean distance between the starting and ending activity locations, in the same units the network tables use.
SEL_USE_FINISH_TRIP	Directs the Collator to include a field that is true if an "end trip" event is found for this trip in the event output file.

Configuration File Key	Description
SEL_USE_MODE_LEG_COUNT	Directs the Collator to include a field giving the number of legs on the trip using the specified mode. The argument is a comma-separated list of modes. Currently, only the following modes are distinguished: w - walk i - bicycle t, l, or b - transit c - non-transit vehicle (as driver or passenger) a - activity
SEL_USE_MODE_STRING	Directs the Collator to include a field reflecting the modes used on every leg of this trip. The value is a string with one letter for each leg, starting from the left. Information comes from the Plan file, which does not know about as many modes as the activity file. Currently the letters used and their meanings are: c - driving a vehicle p - passenger in a non-transit vehicle t - transit w - walk i - bicycle a - activity (no transportation)
SEL_USE_NUMLEGS	Directs the Collator to include a field giving the number of legs in this trip, as found in the Plan file.
SEL_USE_OR	Directs the Collator to include a field (for each pair) which is the logical OR between the values of the two fields. The argument is a semicolon-separated list of comma-separated Iteration Database field names.
SEL_USE_PRODUCT	Directs the Collator to include a field (for each pair) which is the product of the values of the two fields. The argument is a semicolon-separated list of comma-separated Iteration Database field names.
SEL_USE_RATIO	Directs the Collator to include a field (for each pair) which is the ratio between the values of the two fields. The argument is a semicolon-separated list of comma separated Iteration Database field names.
SEL_USE_RELDIFF	Directs the Collator to include a field (for each pair) which is the relative difference between the values of the two fields. The argument is a semicolon-separated list of comma-separated Iteration Database field names. SEL_USE_RELDIFF A, B creates a field with values (A - B) / B.
SEL_USE_START_ACT_USER_DATA	Directs the Collator to include a field giving the value of any user-specified field in the Activity Location network table for the activity location at the beginning of the trip. The argument is a semicolon-separated list of field names.

Configuration File Key	Description
SEL_USE_START_IN_REGION	Adds a field to the Iteration Database which has the value true if the starting activity location is in the specified polygon. The argument is a semicolon-separated list of UAZ region identifiers. Each region identifier is of the form <UAZ_ID>, <polygon id>; where <UAZ_ID> is an integer referring to a User Analysis Zone file specified by the UAZ_FILE_NAME configuration file key and <polygon id> is the ID of a polygon contained within that file. The Iteration Database column will be named START_IN_REGION_<UAZ_ID>_<polygon ID>. Its value will be a boolean, which is true if the starting activity location for the trip is inside the polygon.
SEL_USE_START_REGION	The number (n) of the User Analysis Zone specified by the configuration file key SEL_UAZ_FILE_n. The value of the field in the iteration database is the number of the polygon in the UAZ that contains the starting activity location or –1 if not in any defined polygon.
SEL_USE_SUM	Directs the Collator to include a field (for each pair) which is the sum of the values of the two fields. The argument is a semicolon-separated list of comma-separated Iteration Database field names.
SEL_USE_T_MODE	Directs the Collator to include a field giving the total time (in seconds) spent in a particular mode on this trip, as found from the expected duration of those legs in the Plan file. See SEL_USE_MODE_LEG_COUNT for a list of the modes that can be distinguished. The argument should be a semicolon-separated list of mode characters.
SEL_USE_T_TOTAL	Directs the Collator to include a field giving the total time spent on this trip as found from the difference in the TIME field between the last and first events for this trip. The event output file should capture events with "change on trip" status.
SEL_USE_T_WAIT	Directs the Collator to include a field giving the total time spent waiting as found by summing over the times between "begin waiting" and "end waiting" events in the event output file. The event output file must capture these events and must not filter out the TIME field.
SEL_USE_T_WALK	Directs the Collator to include a field giving the total time (in seconds) spent walking on this trip, as found from the expected duration of walk legs in the Plan file.
SEL_USE_TRAVERSE_REGION	The number (n) of the User Analysis Zone specified by the configuration file key SEL_UAZ_FILE_n. The value of the field in the iteration database is true if one of the start/end activity locations is in the specified UAZ and the other is outside the UAZ.

Configuration File Key	Description
`SEL_USE_XOR`	Directs the Collator to include a field (for each pair) which is the logical exclusive `OR` between the values of the two fields. The argument is a semicolon-separated list of comma-separated Iteration Database Field names.

Chapter Six: Index

Absolute criteria, *2*
ACT_HOME_ACTIVITY_TYPE, *13*
ACT_POPULATION_FILE, *14*
ACT_SCHOOL_ACTIVITY_TYPE, *13*
ACT_WORK_ACTIVITY_TYPE, *13*
Activity file, *1*
Activity files, *9*
Activity Generator, *4*, *6*, *7*, *8*, *12*
Activity location table, *10*
Activity locations, *1*
ACTIVITY_FILE, *13*
Algorithm, *9*, *11*, *12*
Biased selection, *1*
Binning, *11*
Collator, *9*, *10*, *11*
Discretization, *11*
Euclidean distance, *10*
Event files, *9*
Feedback, *1*, *12*
Framework, *2*
Framework flexibility, *7*
Input/Output, *1*
ITDB_USE_CROSS_BOUNDARY, *10*
ITDB_USE_DIFF, *11*
Iteration Database, *1*, *2*, *4*, *6*, *7*, *8*, *9*, *12*
Iteration script, *1*, *4*, *6*, *8*
Iterative feedback, *1*
Iterative process, *4*
Learned behavior, *1*
Mode selections, *1*
NET_ACTIVITY_LOCATION_TABLE, *13*
NET_DIRECTORY, *13*
NET_LINK_TABLE, *13*
NET_NODE_TABLE, *13*
NET_PARKING_TABLE, *13*
NET_PROCESS_LINK_TABLE, *13*
NET_TRANSIT_STOP_TABLE, *13*
Network characteristics, *8*
Output Visualizer, *10*
Plan file, *1*
Plan files, *9*
PLAN_FILE, *13*
Population file, *1*
Population Synthesizer, *9*
Processing algorithms, *10*
Relative criteria, *2*
Route Planner, *1*, *4*, *6*, *7*, *8*
Route planning, *1*
ROUTER_IGNORABLE_PRIORITIES, *13*
SEL_ACTIVITY_FILE, *13*
SEL_ALGORITHM, *12*, *23*
SEL_ALGORITHM_n_PARAMETER_m, *12*
SEL_BI_SEL_ALGO, *12*
SEL_BIN_BOUNDS_n, *23*
SEL_BIN_FIELD_<n>., *11*
SEL_BIN_FIELD_n, *23*
SEL_BIN_NAME_<n>., *11*
SEL_BIN_NAME_n, *23*
SEL_BIN_NUMBINS_n, *23*
SEL_BIN_SEL_ALGO, *23*
SEL_BINS_NUMBINS_n, *23*
SEL_COST, *12*, *23*, *24*
SEL_EVENT_FILE, *13*
SEL_GOAL, *12*, *24*
SEL_HOUSEHOLD_FILE, *13*
SEL_ITDB_FILE, *13*
SEL_MESSAGE_LEVEL, *13*
SEL_NO_ITDB_INDEX, *13*
SEL_PLAN_FILE, *13*
SEL_POPULATION_FILE, *14*
SEL_START_BINS, *12*
SEL_STRAT_BINS, *24*
SEL_STRAT_OUT_FILE, *14*
SEL_UAZ_FILE_n, *14*, *25*, *27*
SEL_USE_<pop_file header field>, *22*
SEL_USE_ACCELS, *18*
SEL_USE_ACT_GROUP_NUM, *15*
SEL_USE_ACT_HH_ID, *15*
SEL_USE_ACT_ID, *15*
SEL_USE_ACT_LAST_OK_ITER, *15*
SEL_USE_ACT_LOCATION, *15*
SEL_USE_ACT_PESON_ID, *15*
SEL_USE_ACT_TYPE, *15*
SEL_USE_AND, *25*
SEL_USE_ANOMALY, *18*
SEL_USE_BLOCK_GROUP, *22*
SEL_USE_COST, *20*
SEL_USE_CROSS_BOUND, *10*, *25*
SEL_USE_DEP_TIME, *20*

SEL_USE_DIFF, *25*
SEL_USE_DISTANCE_SUM, *18*
SEL_USE_DRIVER, *20*
SEL_USE_DRIVES_PASSENGER, *11*, *25*
SEL_USE_DURATION, *20*
SEL_USE_EFFSPEED, *11*, *25*
SEL_USE_END_ACC, *20*
SEL_USE_END_ACC_TYPE, *20*
SEL_USE_END_ACT_USER_DATA, *11*, *25*
SEL_USE_END_DUR_LB, *15*
SEL_USE_END_DUR_UB, *15*
SEL_USE_END_IN_REGION, *10*, *25*
SEL_USE_END_REGION, *25*
SEL_USE_END_TIME_LB, *15*
SEL_USE_END_TIME_UB, *15*
SEL_USE_EUCLID, *10*, *25*
SEL_USE_EVENT_LEG_ID, *18*
SEL_USE_EVENT_PERSON_ID, *18*
SEL_USE_EVENT_TRIP_ID, *18*
SEL_USE_EVENT_USER, *18*
SEL_USE_EVENT_VEHICLE_ID, *18*
SEL_USE_FINISH_TRIP, *11*, *25*
SEL_USE_GCF, *20*
SEL_USE_HH_ID, *22*
SEL_USE_HOME_LOCATION, *22*
SEL_USE_LEG_ID, *20*
SEL_USE_LINK, *18*
SEL_USE_LOCATION, *18*
SEL_USE_MAX_TIME, *20*
SEL_USE_MODE, *21*
SEL_USE_MODE_LEG_COUNT, *11*, *26*, *27*
SEL_USE_MODE_PREF, *16*
SEL_USE_MODE_STRING, *11*, *26*
SEL_USE_NODE, *18*
SEL_USE_NUMBER_PERSONS, *22*
SEL_USE_NUMBER_VEHICLES, *22*
SEL_USE_NUMLEGS, *11*, *26*
SEL_USE_OR, *26*
SEL_USE_OTHER_PARTICIPANTS, *16*
SEL_USE_PERSON_ID, *22*
SEL_USE_PLAN_PERSON_ID, *21*
SEL_USE_PLAN_USER, *21*
SEL_USE_POSS_LOC, *16*
SEL_USE_PRIORITY, *16*
SEL_USE_PRODUCT, *26*
SEL_USE_RATIO, *26*
SEL_USE_RELDIFF, *26*
SEL_USE_ROUTE, *19*
SEL_USE_ROUTER_LAST_OK_ITER, *21*
SEL_USE_SIGNALS, *19*
SEL_USE_START_ACC, *21*
SEL_USE_START_ACC_TYPE, *21*
SEL_USE_START_ACT_GROUP_NUM, *16*
SEL_USE_START_ACT_ID, *16*
SEL_USE_START_ACT_LOCATION, *16*
SEL_USE_START_ACT_TYPE, *16*
SEL_USE_START_ACT_USER_DATA, *11*, *26*
SEL_USE_START_DUR_LB, *16*
SEL_USE_START_DUR_UB, *16*
SEL_USE_START_END_TIME_LB, *17*
SEL_USE_START_END_TIME_UB, *17*
SEL_USE_START_IN_REGION, *10*, *25*, *27*
SEL_USE_START_MODE_PREF, *17*
SEL_USE_START_OTHER_PARTICIPANTS, *17*
SEL_USE_START_POSS_LOC, *17*
SEL_USE_START_PRIORITY, *17*
SEL_USE_START_REGION, *27*
SEL_USE_START_START_TIME_LB, *17*
SEL_USE_START_START_TIME_UB, *17*
SEL_USE_START_TIME_LB, *16*
SEL_USE_START_TIME_UB, *16*
SEL_USE_START_VEHICLE_ID, *17*
SEL_USE_STATUS, *19*
SEL_USE_STOP_TIME, *21*
SEL_USE_STOPPED, *19*
SEL_USE_STOPS, *19*
SEL_USE_STRATIFICATION, *24*
SEL_USE_SUM, *27*
SEL_USE_T_MODE, *11*, *27*
SEL_USE_T_TOTAL, *11*, *25*, *27*
SEL_USE_T_WAIT, *11*, *27*
SEL_USE_T_WALK, *11*, *27*
SEL_USE_TIME, *19*
SEL_USE_TIME_SUM, *19*
SEL_USE_TRACT, *22*
SEL_USE_TRAVERSE_REGION, *27*
SEL_USE_TRIP_ID, *21*
SEL_USE_TURN, *19*
SEL_USE_VEH_SUBTYPE, *19*
SEL_USE_VEH_TYPE, *19*
SEL_USE_VEHICLE_ID, *16*

SEL_USE_XOR, *28*
SEL_USE_YIELDS, *19*
Selection process, *1*, *2*, *3*, *6*
Selector, *1*, *4*, *5*, *6*, *7*, *8*, *9*, *12*
Selector statistics, *7*
Simulated population, *1*
Simulation output file, *1*
Stratifier, *9*, *11*, *12*
TAZ, *10*
Traffic analysis zones, *10*
Traffic information systems, *8*
Traffic Microsimulator, *1*, *6*, *7*, *8*
TRANSIMS Framework, *1*
Transit schedules, *8*
Travel time table, *1*, *2*
Traveler, *1*, *2*, *4*, *6*, *7*, *8*, *12*
UAZ, *10*
UAZ_FILE_n, *10*
Updating travelers, *1*
User analysis zones, *10*
Vehicle file, *1*